CHAPTER 11

Calculation of The Flow Field

Outline

11-1 Need for a Special Procedure

- The real difficulty in the calculation of the velocity field lies in the unknown pressure field. The pressure gradient forms a part of source term for a momentum equation.

11-2 Some Related Difficulties (1)

- Representation of the Pressure-Gradient Term

1. To integrate $-dp/dx$ over the control volume shown in Fig.1, we can get p_w-p_e.

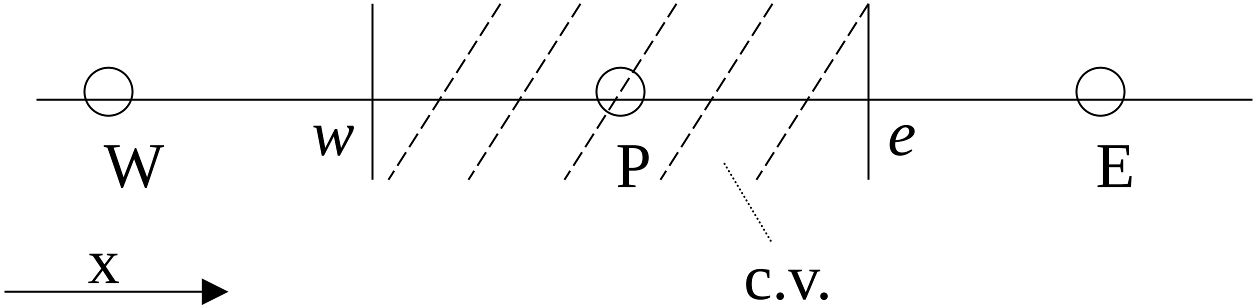


Fig. 1

11-2 Some Related Difficulties (2)

2. To express p_w-p_e in terms of the grid-point pressure, we may assume a piecewise-linear profile for pressure. Therefore, we can get

$$p_w - p_e = \frac{p_W + p_p}{2} - \frac{p_p + p_E}{2} = \frac{p_W - p_E}{2}$$

This means that the momentum equation will contain the pressure difference between two alternate grid points, and not between adjacent ones.

11-2 Some Related Difficulties (3)

3. There is another implication that is far more serious. It can be seen from Fig.2, where a pressure field is proposed in terms of the grid-point values of pressure.

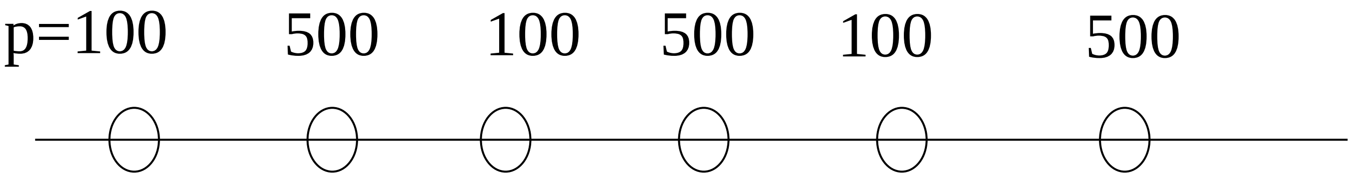


Fig. 2

such a zig-zag field cannot be regarded as realistic; but for any grid point p, the corresponding p_W-p_E can be seen to zero, since the alternate pressure values are everywhere equal.

11-2 Some Related Difficulties (4)

- Representation of the continuity Equation

  If we integrate the continuity equation over the c.v. shown in Fig1, we have

$$u_e - u_w = 0$$

11-2 Some Related Difficulties (5)

- Once again, the use of a piecewise-linear profile for u and of the midway locations of the control-volume faces leads to

$$\frac{u_p - u_E}{2} - \frac{u_W - u_p}{2} = 0 \qquad or \qquad u_E - u_W = 0$$

Thus, the discretized continuity equation demands the quality of velocities at alternate grid points and not at adjacent ones.

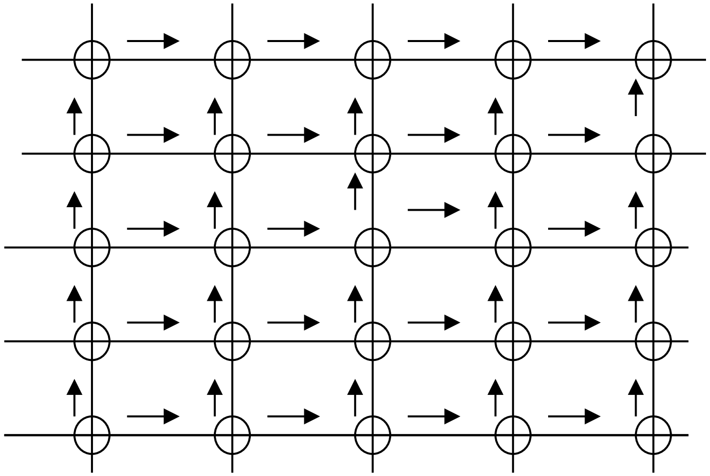
11-3 A Remedy : The staggered Grid (1)

- The difficulties described so far can be resolved by recognizing that we do not have to calculate all the variables for the same grid points. We can, if we wish, employ a different grid for each dependent variable.

- Staggered grid:
  1. The velocity components are calculated for the points that lie on the face of the control volume, as shown in Fig.3.
  2. Other variables are calculated for the grid points(small circles).

Fig.3 $\rightarrow = u_{i,}$, $\uparrow = v_i$,

○ =other variables



11-3 A Remedy : The staggered Grid (3)

3. The important advantages are twofold:
   a) For a typical c.v. shown in Fig.3, it is easy to see that the discretized continuity equation would contain the difference of adjacent velocity components, and this would prevent a wavy velocity field.
   b) The second important advantage of the staggered grid is that the pressure difference between two adjacent grid points now becomes the natural driving force for the velocity component located between these grid points.

11-4 The Momentum Equations (1)

- The discretization equation (2-D) can be written as

$$a_e u_e = \sum a_{nb} u_{nb} + b + \left(p_P - p_E\right) A_e \quad ---(1a)$$

$$a_n v_n = \sum a_{nb} v_{nb} + b + \left(p_P - p_N\right) A_n \quad ---(1b)$$

$$similarly \quad a_t w_t = \sum a_{nb} w_{nb} + b + \left(p_P - p_t\right) A_t \quad ---(1c)$$

11-4 The Momentum Equations (2)

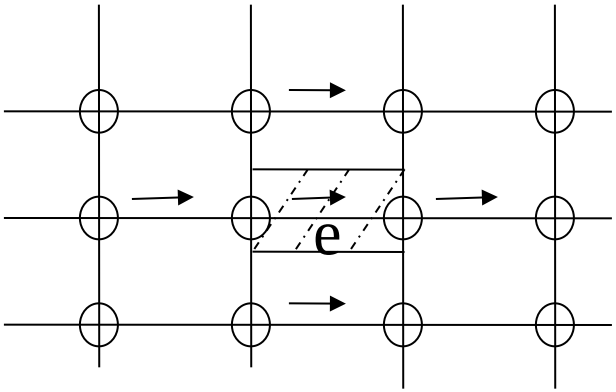


Fig.4a c.v. for u

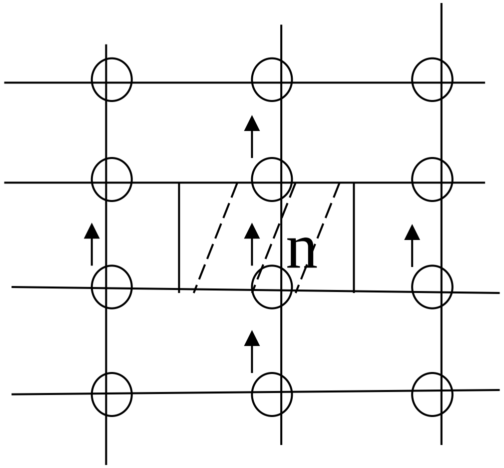


Fig.4b c.v. for v

The pressure gradient is not included in the source-term S_C and S_P.

11-4 The Momentum Equations (3)

- The momentum equation can be solved when the pressure field is given or is somewhat estimated. Unless the correct pressure field is employed, the resulting velocity field will not satisfy the continuity equation. Such an imperfect field based on a guessed pressure field p^* will be denoted by u^*,v^*,w^*. This "starred" velocity field will result from the solution of the following equations:

$$a_e u_e^* = \sum a_{nb} u_{nb}^* + b + \left(p_P^* - p_E^*\right) A_e \quad ---(2a)$$

$$a_n v_n^* = \sum a_{nb} v_{nb}^* + b + \left(p_P^* - p_N^*\right) A_n \quad ---(2b)$$

$$a_t w_t^* = \sum a_{nb} w_{nb}^* + b + \left(p_P^* - p_t^*\right) A_t \quad ---(2c)$$

11-5 The Pressure and Velocity Corrections (1)

- One aim is to find a way of improving the guessed pressure p* such that the resulting starred velocity field will progressively get closer to satisfying the continuity equation. Let us propose that the correct pressure p is obtained from

$$p = p^* + p' \quad ---(3)$$

where *p'* will be called the pressure correction.

11-5 The Pressure and Velocity Corrections (2)

- Similarly, we can get

$$u = u^* + u' \quad ---(4a),$$

$$v = v^* + v' \quad ---(4b),$$

$$w = w^* + w' \quad ---(4c),$$

$$u', v', w' = velocity \quad corrections$$

- If (1a)-(2a), we have

$$a_e u'_e = \sum a_{nb} u'_{nb} + \left(p'_p - p'_E\right) A_e \quad ---(5)$$

At this point, we shall boldly decide to drop the term $\sum a_{nb} u'_{nb}$ from eq.(5) and the result is

$$a_e u'_e = \left(p'_p - p'_E\right) \quad or \quad u'_e = d_e\left(p'_p - p'_E\right) \quad --(6)$$

$$where \quad d_e = \frac{A_e}{a_e}$$

- Eq.(6) will be called the velocity-correction formula, which can also be written as

$$u_e = u_e^* + d_e\left(p_p' - p_E'\right) \quad --(7a)$$

$$Similarly, \quad v_n = v_n^* + d_n\left(p_p' - p_n'\right) \quad --(7b)$$

$$w_t = w_t^* + d_t\left(p_p' - p_t'\right) \quad --(7c)$$

- The discretization eq. of continuity eq:

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}+\frac{\partial(\rho w)}{\partial z}=0$$

$$\frac{\left(\rho_p-\rho_p^0\right)\Delta x\Delta y\Delta z}{\Delta t}+\left[(\rho u)_e-(\rho u)_w\right]\Delta y\Delta z$$

$$+\left[(\rho v)_n-(\rho v)_s\right]\Delta z\Delta x+\left[(\rho w)_t-(\rho w)_b\right]\Delta x\Delta y=0 \quad ---(8)$$

11-6 The Pressure-Correction Equation (2)

- Substituting eq. (7) into eq. (8), we can obtain

$$a_p p'_p = a_E p'_E + a_W p'_W + a_N p'_N + a_S p'_S + a_T p'_T + a_B p'_B + b \quad --(9)$$

$$\begin{aligned} where \quad & a_E = \rho_e d_e \Delta y \Delta z, \quad a_W = \rho_w d_w \Delta y \Delta z, \\ & a_N = \rho_n d_n \Delta y \Delta z, \quad a_S = \rho_s d_s \Delta y \Delta z, \\ & a_T = \rho_t d_t \Delta y \Delta z, \quad a_B = \rho_b d_b \Delta y \Delta z, \\ & a_p = a_E + a_W + a_N + a_S + a_T + a_B \\ & b = \frac{\left(\rho_p^0 - \rho_p\right)\Delta x \Delta y \Delta z}{\Delta t} + \left[\left(\rho u^*\right)_w - \left(\rho u^*\right)_e\right]\Delta y \Delta z \\ & \quad + \left[\left(\rho v^*\right)_s - \left(\rho v^*\right)_n\right]\Delta z \Delta x + \left[\left(\rho w^*\right)_b - \left(\rho w^*\right)_t\right]\Delta x \Delta y \quad ---(8) \end{aligned}$$

11-6 The Pressure-Correction Equation (3)

- If b is zero, it means that the stained velocity, in conjunction with the available value of (ρ_p^0- ρ_p), do satisfy the continuity equation and no pressure correction is needed. The term b this represents a " mass source", which the pressure corrections must be annihilate.

11-7 The SIMPLE Algorithm (1)

- SIMPLE stands for <u>S</u>emi-<u>I</u>mplicit <u>M</u>ethod for <u>P</u>ressure-<u>L</u>inked <u>E</u>quations.
- Sequence of operations:
  1. Guess the pressure p^*.
  2. Solve the momentum eqs, such as (2a)~(2c), to obtain u^*, v^*, w^*.
  3. Solve p' eq.
  4. Calculate $p=p^*+p'$
  5. Calculate u,v,w from (7a)~(7c)
  6. Solve other Φ's.
  7. Treat the corrected pressure p as a new guessed pressure p^*, return to step (2) and repeat the whole procedure until a converged solution is obtain.

11-7 The SIMPLE Algorithm (2)

- Discussion of the Pressure-Correction Equation
  1. If expressions such as $a_{nb}u'_{nb}$ were retained, they would have to be expressed in terms of the pressure corrections and the velocity corrections at the neighbors of u_{nb}. The omission of the $\sum a_{nb}u'_{nb}$ term enables us to cast the p equation in the same form as the general Φ equation, and to adopt a sequential, one-variable-at-a-time, solution procedure.

11-7 The SIMPLE Algorithm (3)

2. The words semi-implicit in the name SMPLE have been used to acknowledge the omission of the term $\sum a_{nb}u'_{nb}$. This term represents on indirect or implicit influence of the pressure correction on velocity. Pressure corrections at nearby locations can alter the neighboring velocities and thus a velocity correction at the point under consideration. We do not include this influence and thus work with a scheme that is only partially, and not totally, implicit.

11-7 The SIMPLE Algorithm (4)

3. The omission of any term, would of course, be unacceptable if it meant that the ultimate solution would not be true solution of the discretized forms of the momentum and continuity equation. It also happens that the converged solution given by SIMPLE does not contain any error resulting from the omission of $\sum a_{nb}u'_{nb}$

11-7 The SIMPLE Algorithm (5)

4. The mass source b serves as a useful indicator of the convergence of the fluid-flow solution. The iterations should be continued until the value of b everywhere becomes sufficiently small.
5. The pressure-correction can be seen to be merely an intermediate algorithm that leads us to the correct pressure field, but has no direct effect on the final solution.

11-7 The SIMPLE Algorithm (6)

6. The pressure-correction equation is prone to divergence unless some under-relaxation is used. A generally successful practice can be described as follows: we under-relax u^*, v^*, w^* while solving the momentum equations (with a relaxation factor α=0.5). Also, we employ

$$p = p^* + \alpha_p p' \quad , \quad \alpha_p = 0.8$$

It is not implied that these values are the optimum ones or will even produce divergence for all problems. The optimum relaxation factor values are usually problem-dependent.

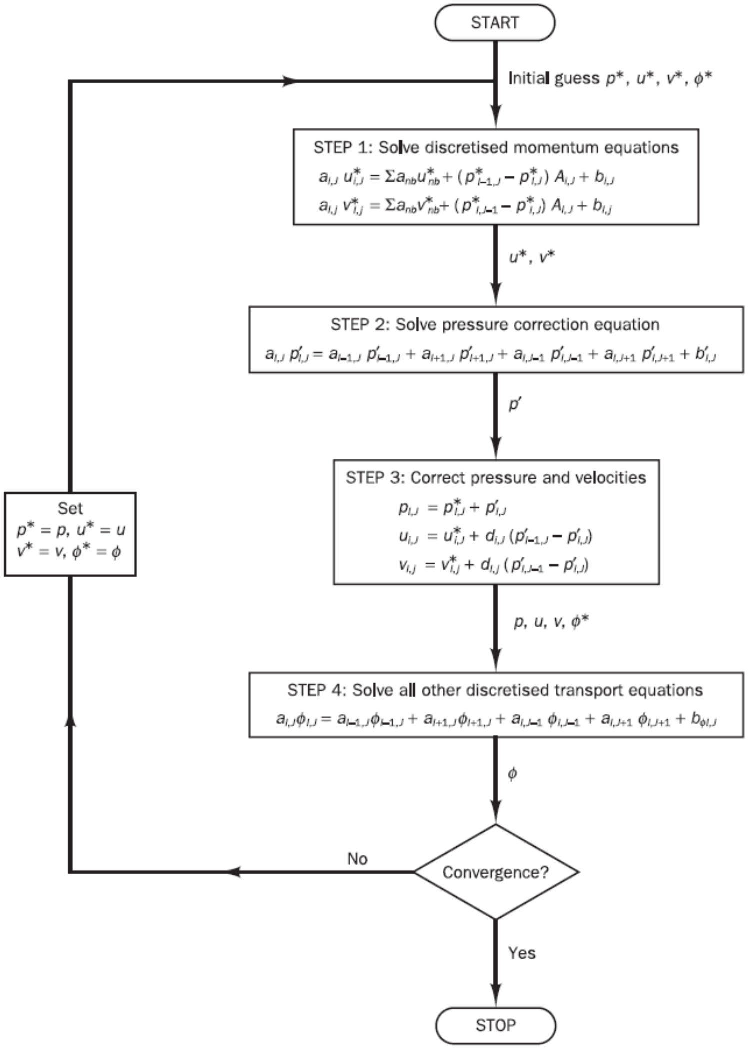
11-7 The SIMPLE Algorithm (7)

- The Relative Nature of Pressure
  1. Since no boundary pressure is specified and all the boundary coefficients such as a_E will be zero, the p' equation is left without any means of establishing the absolute value of p'. The coefficients of the p' equation are such that $a_P = \sum a_{nb}$; this means that p' and p' +c (c is an arbitrary constant) would both satisfy the p' equation.

11-7 The SIMPLE Algorithm (8)

2. Only difference in the pressure are meaningful ($p=p^*+p'$), and these are not altered by an arbitrary constant to the p' field. Pressure is then a relative variable, not a absolute one.

11-7 The SIMPLE Algorithm (9)



11-8 The SIMPLEC Algorithm

- SIMPLEC stands for SIMPLE-Consistent.
- It follows the same steps as the SIMPLE algorithm.
- The u-velocity correction equation of SIMPLEC is given by

$$u_e = d_e\left(p'_e - p'_e\right) \quad , \quad d_e = \frac{A_e}{a_e - \sum a_{nb}}$$

$$\left(note : a_e u'_e = \sum a_{nb} u'_{nb} + \left(p'_p - p'_E\right) A_e\right)$$

11-9 The PISO Algorithm (1)

The PISO algorithm, which stands for Pressure Implicit with Splitting of Operators, of Issa (1986) is a pressure–velocity calculation procedure developed originally for non-iterative computation of unsteady compressible flows. It has been adapted successfully for the iterative solution of steady state problems. PISO involves one predictor step and two corrector steps and may be seen as an extension of SIMPLE, with **a further corrector step** to enhance it.

Predictor step

Discretised momentum equations Eqs. (2a)-2(c) are solved with a guessed or intermediate pressure field p^* to give velocity components u^* and v^* using the same method as the SIMPLE algorithm.

11-9 The PISO Algorithm (2)

Corrector step 1

The u^* and v^* fields will not satisfy continuity unless the pressure field p^* is correct. The first corrector step of SIMPLE is introduced to give a velocity field (u^{**}, v^{**}) which satisfies the discretised continuity equation. The resulting equations are the same as the velocity correction equations Such as Eq. (6) of SIMPLE but, since there is a further correction step in the PISO algorithm, we use a slightly different notation:

$$p^{**} = p^* + p'$$
$$u^{**} = u^* + u'$$
$$v^{**} = v^* + v'$$

These formulae are used to define corrected velocities u^{**} and v^{**}:

$$u^{**}_{i,J} = u^*_{i,J} + d_{i,J}(p'_{I-1,J} - p'_{I,J}) \tag{6.50}$$

$$v^{**}_{I,j} = v^*_{I,j} + d_{I,j}(p'_{I,J-1} - p'_{I,J}) \tag{6.51}$$

11-9 The PISO Algorithm (3)

As in the SIMPLE algorithm equations (6.50)–(6.51) are substituted into the discretised continuity equation to yield pressure correction equation Eq. (9).

In PISO, Eq. (9) is called the first pressure correction equation.

It is solved to yield the first pressure correction field p'. Once the pressure corrections are known, the velocity components u^{**} and v^{**} can be obtained through equations (6.50)–(6.51).

11-9 The PISO Algorithm (4)

Corrector step 2

To enhance the SIMPLE procedure PISO performs a second corrector step. The discretised momentum equations for u^{**} and v^{**} are

$$a_{i,J}u^{**}_{i,J} = \sum a_{nb}u^{*}_{nb} + (p^{**}_{I-1,J} - p^{**}_{I,J})A_{i,J} + b_{i,J} \quad (6.12)$$

$$a_{I,j}v^{**}_{I,j} = \sum a_{nb}v^{*}_{nb} + (p^{**}_{I,J-1} - p^{**}_{I,J})A_{I,j} + b_{I,j} \quad (6.13)$$

A twice-corrected velocity field (u^{***}, v^{***}) may be obtained by solving the momentum equations once more:

$$a_{i,J}u^{***}_{i,J} = \sum a_{nb}u^{**}_{nb} + (p^{***}_{I-1,J} - p^{***}_{I,J})A_{i,J} + b_{i,J} \quad (6.52)$$

$$a_{I,j}v^{***}_{I,j} = \sum a_{nb}v^{**}_{nb} + (p^{***}_{I,J-1} - p^{***}_{I,J})A_{I,j} + b_{I,j} \quad (6.53)$$

Note that the summation terms are evaluated using the velocities u^{**} and v^{**} calculated in the previous step.

Subtraction of equation (6.12) from (6.52) and (6.13) from (6.53) gives

$$\boxed{u^{***}_{i,J} = u^{**}_{i,J} + \frac{\sum a_{nb}(u^{**}_{nb} - u^{*}_{nb})}{a_{i,J}} + d_{i,J}(p''_{I-1,J} - p''_{I,J})} \quad (6.54)$$

$$\boxed{v^{***}_{I,J} = v^{**}_{I,j} + \frac{\sum a_{nb}(v^{**}_{nb} - v^{*}_{nb})}{a_{I,j}} + d_{I,j}(p''_{I,J-1} - p''_{I,J})} \quad (6.55)$$

where p'' is the second pressure correction so that p^{***} may be obtained by

$$p^{***} = p^{**} + p'' \quad (6.56)$$

Substitution of u^{***} and v^{***} in the discretised continuity equation (6.29) yields a second pressure correction equation

$$\boxed{a_{I,J}p''_{I,J} = a_{I+1,J}p''_{I+1,J} + a_{I-1,J}p''_{I-1,J} + a_{I,J+1}p''_{I,J+1} + a_{I,J-1}p''_{I,J-1} + b''_{I,J}} \quad (6.57)$$

11-9 The PISO Algorithm (5)

with $a_{I,J} = a_{I+1,J} + a_{I-1,J} + a_{I,J+1} + a_{I,J-1}$, and the neighbour coefficients are as follows:

$a_{I+1,J}$	$a_{I-1,J}$	$a_{I,J+1}$	$a_{I,J-1}$	$b''_{I,J}$
$(\rho dA)_{i+1,J}$	$(\rho dA)_{i,J}$	$(\rho dA)_{I,j+1}$	$(\rho dA)_{I,j}$	$\left[\left(\frac{\rho A}{a}\right)_{i,J}\sum a_{nb}(u^{**}_{nb} - u^{*}_{nb}) - \left(\frac{\rho A}{a}\right)_{i+1,J}\sum a_{nb}(u^{**}_{nb} - u^{*}_{nb}) + \left(\frac{\rho A}{a}\right)_{I,j}\sum a_{nb}(v^{**}_{nb} - v^{*}_{nb}) - \left(\frac{\rho A}{a}\right)_{I,j+1}\sum a_{nb}(v^{**}_{nb} - v^{*}_{nb})\right]$

In the derivation of (6.57) the source term

$$[(\rho A u^{**})_{i,J} - (\rho A u^{**})_{i+1,J} + (\rho A v^{**})_{I,j} - (\rho A v^{**})_{I,j+1}]$$

is zero since the velocity components u^{**} and v^{**} satisfy continuity.

11-9 The PISO Algorithm (6)

Equation (6.57) is solved to obtain the second pressure correction field p'', and the twice-corrected pressure field is obtained from

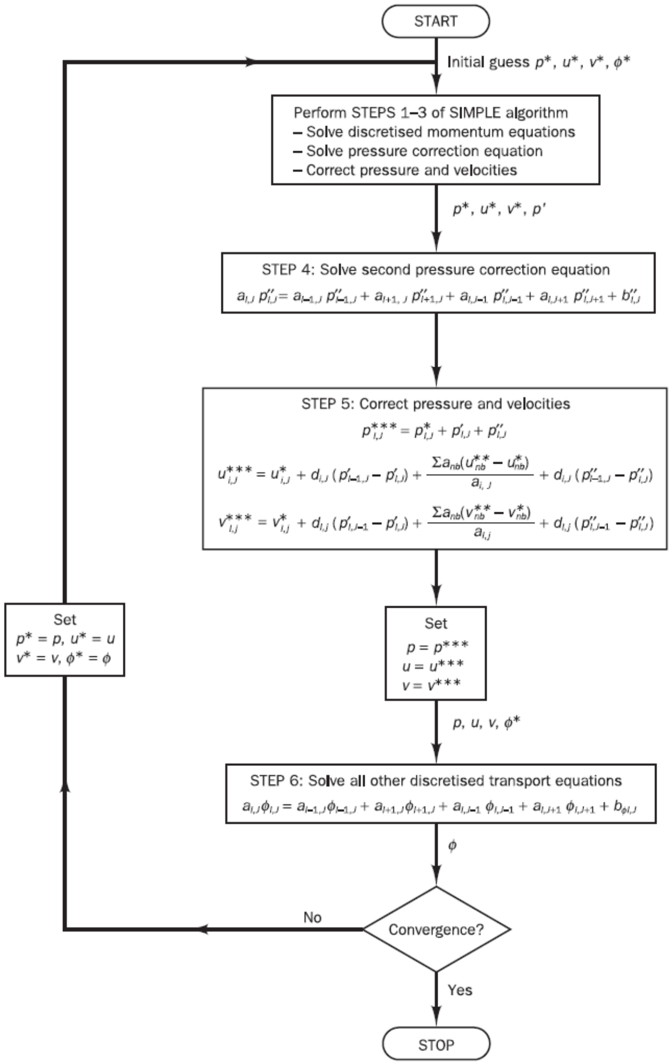
$$\boxed{p^{***} = p^{**} + p'' = p^{*} + p' + p''} \tag{6.58}$$

Finally the twice-corrected velocity field is obtained from equations (6.54)–(6.55).

In the non-iterative calculation of unsteady flows the pressure field p^{***} and the velocity field u^{***} and v^{***} are considered to be the correct u, v and p. The sequence of operations for an iterative steady state PISO calculation is given in Figure 6.8.

11-9 The PISO Algorithm (7)

Figure 6.8 The PISO algorithm



11-9 The PISO Algorithm (8)

The PISO algorithm solves the pressure correction equation twice so the method requires additional storage for calculating the source term of the second pressure correction equation. As before, under-relaxation is required with the above procedure to stabilise the calculation process. Although this method implies a considerable increase in computational effort it has been found to be efficient and fast. For example, for a benchmark laminar backward-facing step problem Issa *et al.* (1986) reported a reduction of CPU time by a factor of 2 compared with standard SIMPLE.

11-10 Convergence Criterion

- $a_p \Phi_p = \sum a_{nb} \Phi_{nb} + b$
- Residual $R = \sum a_{nb} \Phi_{nb} + b - a_p \Phi_p$
- Obviously, when the discretization eq is satisfied, R will be zero. A suitable convergence criterion is to require that the largest value of |R| be less than a certain small number.